SHAPE

the SHarp Application Programming Environment

TUTORIAL

Helsingør, 1987-8-6

Copyright ©mcmlxxxvii I.P.Sharp ApS

SHAPE Tutorial

This TUTORIAL quickly takes you through the process of creating a small "Personnel System", which contains Employee Records. The Data is stored in a VIEWPOINT file. The "application" consists of a single Input Form from which you can enter employee data. We imagine that you will use VIEWPOINT to create Reports for the system.

It may not be possible to implement everything described in this tutorial, but we hope that it will give a reasonable idea of what it will be like to use SHAPE to create small application. Let us start with the SHAPE Main Menu. We are signed on to an account with ALIAS 'SHAPE':

```
---------- System:  SHAPE.                    ----------

---------- Shape 0.0 ----------

                    WELCOME TO SHAPE

   DOCUMENT   Produce Documentation.     DATA     Define DATA types.

   SYSTEMS    System Definitions.        FILES    Describe FILES/DATASETS.

   MENUS      Define Menus.              SCREENS  Create Screens.

   PROFILE    Edit User Profile.         APL      Maintain APL-code.

   MACROS     User-defined Macros.       ADMIN    System Administrator.

              To select an option, use the tab key to move the
              cursor opposite the option name, or type the name
              of the option on the Command Line, and press ENTER.

---------- 1=Help, 3=Exit ----------
```

Screen 1: SHAPE Main Menu

The first thing we have to do is to create a SYSTEM. Type SYSTEMS on the command line at the top of the menu screen, and press enter.

Enter the name of the new system, and a short description, as follows:

```
----------------- System SHAPE.                                  --------------

System   1 of  1                                  Systems in Library: SHAPE
---------------------------------- SYSTEMS -------------------------------------

  >System             >Description
   TUTORIAL            Tutorial System




---- 1=Help, 2=Send, 3=Exit, 4=Add, 5=Delete, 6=Zoom, 7=Up, 8=Down, 9=Save ----
```

Screen 2: Creating a System

Press PF9 (SAVE) and the new system will be created. The new thing we have to do is to create a MENU under which to hang our Update Screen. Type 'SYSTEM TUTORIAL;MENUS' on the command line and press ENTER. This will select the newly created system and take us to the Menu Definition screen.

Newly-created systems always have one menu called 'MAIN':

```
------------------ System SHAPE.TUTORIAL: Tutorial System --------------------

Menu  1 of  1
---------------------------------- MENUS -----------------------------------

  >Menu              >Description
   MAIN               TUTORIAL - MAIN MENU




---- 1=Help, 2=Send, 3=Exit, 4=Add, 5=Delete, 6=Zoom, 7=Up, 8=Down, 9=Save ----
```

Screen 3: Menu Overview

Lets just hang our screen under the Main Menu for starters. Tab down to the line on which the menu is defined, and pres PF6 (ZOOM). Alternatively, type the command ZOOM MENU=MAIN and press ENTER.

Enter the line defining our Screen as the first option in the Main Menu:

```
----------------- System SHAPE.TUTORIAL: Tutorial System --------------------

------------------------------ Define Menu ---------------------------------
                                                          Local Objects
  >Menu            >Description                        >Scrn >Cmds  >APL >Mods
   MAIN             TUTORIAL - MAIN MENU                  0     0     0     0

Entry:
Exit.:
---------------------- Options ---------------------------- Option  1 of  1
  N    Type  >Definition          Description                      >Acc  >Hlp
  1      S    EMPFORM             Employee Form                    NO    NO




---- 1=Help, 2=Send, 3=Exit, 4=Add, 5=Delete, 6=Zoom, 7=Up, 8=Down, 9=Save ----
```

Screen 4: Creating a Screen

The four zeros at the top left of the screen tell us that this Menu has no "Local" Screens, Commands, APL objects or Code Modules. The two "NO"s at the end of the line on which you entered the screen name indicate that there is NO access control defined for the new option, nor is there a Help-message for it.

Press PF9 (SAVE) to create the new screen.

When creating a new screen, the system copies a "Site Default" empty screen, which contains a few fields which are supposed to suggest that you should create screens which look like everyone elses. Lets ZOOM on our screen (go into Screen Design mode) to take a look at it:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
-------------------------------- Employee Form -------------------------------


CMD> ____________________________________________________________  mode=EDIT
```

Screen 5: A New Screen

The first and last lines on the screen belong to the Screen Editor. Since the first and last lines of screens are likely to be used for some standard purpose, this should not be too much of a problem (and there is a command to make them go away anyway). In fact, you can claim more lines at the bottom of the screen for the Screen Editor, by issuing the 'PFKEYS' command, which brings one or two PF-key lines at the bottom of the screen, depending on whether you are using 12 or 24 PF-keys). You can move your window on the screen being defined using PF7(UP) and PF8(DOWN)

The two fields '/COMMAND' and '/MESSAGE' are so-called 'special fields', which SHAPE will handle for you while the application is running (they work the same way as the Command and Message lines in Viewpoint or in SHAPE itself). The next line is a '/SCREENNAME' field, which displays the name of the screen, nicely centered on a line of box-drawing characters. We want to add some new fields. Press PF12 (mode) to switch into Quick Definition (QDEF) mode In QDEF mode, existing fields are protected (output only), but you are free to write everywhere else on the screen. Fill in the screen

as follows:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM  [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
------------------------------- Employee Form -------------------------------

        Last Name: _____________________________   Home Phone:_555-5555
       First Name: _____________________________
          Initial(s): _____


           S.I.N.: ___________
      Employee No:_EMPNO___
    Date of Employ:_DDMMMYY


        Address:>1

                                                                  1<

        Comments:>>2

                                                                      2<

CMD> ____________________________________________________________  mode=EDIT
```

Screen 6: Defining some New Fields

"Fixed Text" can be entered as you please, anywhere on the screen. To create a field which is to be used by the programmes for Input or Ouput, you must prefix the field with an underscore. To create a multi-line field, enter a '¿' to the left of the top right of the field, an identification character (almost any character) in the top left and bottom right corners, and a closing '¡' to mark the end. So-called 'wrapped' fields are indicated by a double '¿¿' at the start of the field.

One-line character fields can be entered as a line of underscores, without a leading underscore. Within the field, you can enter the DATA TYPE of the field (for example 'DDMMMYY', or a Format by Example expression (such as 555-5555). SHAPE includes a large number of Data Types, and you are also free to create your own. To demonstrate the creation of a new Data Type, we will define a new 'EMPNO' Data Type for this application.

Once you have completed Quick Definition of new fields, press PF12 again to get back to 'EDIT' mode. The display changes, filling out the multi-row fields you have defined with underscore characters, so you can more easily see the extent of each field. As you can see, some of the fields need a little adjustment of the screen is to be perfect:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM  [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
-------------------------------- Employee Form -------------------------------

        Last Name: ____________________________    Home Phone:_555-5555
       First Name: ____________________________
          Initial(s): _____


            S.I.N.: ___________
       Employee No:_EMPNO__
    Date of Employ:_DDMMMYY


         Address:>_____________________________________________
                  _____________________________________________
                  _____________________________________________<
         Comments:>_____________________________________________________________
--------------------------------------------------------------------------------
________________________________________________________________________________<

CMD> ____________________________________________________________  mode=EDIT
```

Screen 7: An Imperfect Screen

We should make the following corrections:

1. The 'Initials' field should be moved left three positions.
2. It would be nice to have a blank line between the 'Address' and 'Comments' fields.
3. The 'Comments' input field should be extended one position to the left to start in the same column as the Address field.

We're going to make all the changes at once. Type in the following changes on the screen (look at all the exclamation marks to see which changes we mean):

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
-------------------------------- Employee Form -------------------------------

        Last Name: _____________________________    Home Phone:_555-5555
       First Name: _____________________________
          !L3tial(s): !____


           S.I.N.: ___________
      Employee No:_EMPNO__
    Date of Employ:_DDMMMYY


         !U1ress:>!_____________________________________________
                  _____________________________________________
                  _____________________________________________<
         !L1ments:>!L+1_____________________________________________________
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------<

CMD> ____________________________________________________________  mode=EDIT
```

Screen 8: Moving and Extending Fields

The character '!' as the first character entered in a field is a command escape character. We have used the following commands to make the three changes mentioned before:

1. The command 'L3' moves a field Left 3 positions. A '!' alone means 'repeat last command'.
2. 'U1' moves a field Up one line
3. 'L1' moves a field Left 1 position, and 'L+1' command EXTENDS a field to the left by one column.

There are many variations of these commands, allowing you to move or extend fields until they have a given size, until they run into another field, and so on. You could also enter the commands on the Command Line one at a time, and point at the fields you want to affect with the cursor (or enter a single '!' into each field if there are several).

The screen should now look like this:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM  [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
-------------------------------- Employee Form -------------------------------

        Last Name: ____________________________    Home Phone:_555-5555
       First Name: ____________________________
       Initial(s): _____


           S.I.N.: ___________
      Employee No:_EMPNO__
    Date of Employ:_DDMMMYY


         Address:>_____________________________________________
                  _____________________________________________
                  _____________________________________________<

         Comments>_____________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________<

CMD> ZOOM ALL_____________________________________________________  mode=EDIT
```

Screen 9: A Perfect Screen

Now the screen looks the way we want it to, the time has come to adjust other attributes of the fields - the *visible* attributes such as colour, highlighting, and the *invisible* ones such as Data Type, Validation Rules, and so on. There are commands like the ones above to do all these things, for example '!=RED' changes the colour to RED, but there is an easier way for beginners to get going without reading the manual.

Type the command 'ZOOM ALL' on the command line at the bottom of the screen, and press ENTER:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
-------------------------------- Employee Form -------------------------------

       Last Name: ****************************    Home Phone:_555-5555
      First Name: _____________________________
      Initial(s): _____


          S.I.N.: ___________
     Employee No:_EMPNO__
  Date of Employ:_DDMMMYY


       Address:>_____________________________________________
                _____________________________________________
                _____________________________________________<

+--- Field 1 of 9 ------------ Edit Attributes -----------------------------+
!Name: LASTNAME                                 Position: 6 19   Size: 1 30   !
+--Type----Colour--Hilite--Inten--Skip--Read----Write--+>Data  Type-->Validate-!
!  INPUT   GREEN   NONE    NONE   YES   BLANKS  NULLS  ! CHAR        NO       !
+---------------------- 1=Help 3=Back 6=Zoom 12=Next -+----------------------+
```

Screen 10: ZOOM ALL

The ZOOM ALL command starts a sequence which runs through all the fields on the screen except the fixed-text fields, and lets you change the attributes for each one. If you do not want to make a change to a particular field, just press PF12 (NEXT) to skip to the next one.

The ZOOM command puts a little overlay on the screen (it tries to avoid the field you are defining at any given time, but this may not always be possible). Here you can redefine all attributes of the field. For example, in the above screen we have entered the name 'LASTNAME' for the first input field.

You can enter this "attribute editing mode" for one screen at a time rather than running through them all, by pointing at the screen with the cursor (or marking the fields you wish to edit with '!' characters), and pressing PF6 (ZOOM).

After a while we get to the sixth field - Employee Number. This field has the new Data Type (EMPNO) which we have to define:

```
-- SHAPE.TUTORIAL.MAIN.EMPFORM [0] ---------------- Ln  2 to 21, Col  1 to  79
/COMMAND
/MESSAGE
--------------------------------- Employee Form -------------------------------

       Last Name: _____________________________    Home Phone:_555-5555
      First Name: _____________________________
      Initial(s): _____


         S.I.N.: ___________
    Employee No:_*******
  Date of Employ:_DDMMMYY


       Address:>_____________________________________________
                _____________________________________________
                _____________________________________________<

+--- Field 6 of 9 ------------ Edit Attributes ----------------------------+
!Name: EMPNO                                  Position: 12 19  Size: 1 7    !
+--Type----Colour--Hilite--Inten--Skip--Read----Write--+>Data  Type-->Validate-!
!  INPUT   GREEN   NONE    NONE   YES   BLANKS  NULLS  ! EMPNO       NO       !
+---------------------- 1=Help 3=Back 6=Zoom 12=Next -+----------------------+
```

Screen 11: ZOOM on Field "EMPNO"

Tab to the 'Data Type' field, and press PF6 (ZOOM) to take a closer look.

The system has guessed what it could about the Data Type based on the way you entered it on the screen:

```
---------------------- System SHAPE.TUT: Shape Tutorial ---------------------

--------------------------- Edit Data Type: EMPNO [0] -------------------------
  >Type         >Parameters          >Description
   EMPNO

  >Base Data Type: CHARACTER          Internal Data Type: 7A1
                                      Screen Field Width: 7
---- >Input Transforms (Before) -------------------------- Line   1 of   0 ---

---- >Validation (After)  ------------------------------- Line   1 of   1 ---

---- >Output Transforms (After)  ------------------------ Line   1 of   0 ---

---- Validation Example. -------------------------------------------------
Environment:
Input...:
Internal:
Output..:
-- 1=Help 2=Send 3=Back 4=Add 5=Del 6=Zoom 7=Up 8=Dn 9=Save 10=Expand 12=Run --
```

Screen 12: Data Type "EMPNO"

A "Data Type" is based on a so-called 'Base Data Type', which can be ANY other Data Type (there are a handful of Primitive Data types such as Character, Integer and so on which are not based on anything else).

In addition to whatever rules are already in effect for the Base Data Type, you can specify a number of 'Input Transformations' (such as conversion to UpperCase) to be performed BEFORE those of the Base Data Type, and a set of Validation Expressions and Output Transformations to be performed AFTER those of the BDT.

Let us change the Base Data Type to be a defined by a Format By Example expression, and add a single validation expression to check the last digit of the input (and an error message to be issued if the validation fails):

```
------------------- System SHAPE.TUTORIAL: Shape Tutorial -------------------

--------------------------- Edit Data Type: EMPNO [0] ------------------------
  >Type         >Parameters          >Description
   EMPNO                             Employee Number

  >Base Data Type: 5/555-5                        Internal Data Type: I5
                                                  Screen Field Width: 7
---- >Input Transforms (Before) --------------------------- Line   1 of   0 ---


---- >Validation (After)  ---------------------------------- Line   1 of   1 ---
Expr: (10 #residue &DATA) #elementof 1 3 5
Mesg: Last Digit is not one of 1,3,5
---- >Output Transforms (After)  ------------------------- Line   1 of   0 ---


---- Validation Example. ------------------------------------------------------
Environment:
Input...:
Internal:
Output..:
-- 1=Help 2=Send 3=Back 4=Add 5=Del 6=Zoom 7=Up 8=Dn 9=Save 10=Expand 12=Run --
```

Screen 13: The Redefined "EMPNO" Data Type

Validation expressions can be defined on & DATA which means the data in its internal form, after Input Transformations, or on & INPUT, the original user input. Once we enter the new Base Data Type, the Internal Data Type (the Primitive Data Type at the bottom of everything) will be updated (it is used if you want to define Files using the Data Type - as we will do a little later).

Now you can 'test' your data type by entering some input on in the 'Validation Example' section at the bottom of the screen, and pressing PF12 (RUN):

```
-------------------- System SHAPE.TUTORIAL: Shape Tutorial --------------------


-------------------------- Edit Data Type: EMPNO [0] -------------------------
  >Type         >Parameters          >Description
   EMPNO                             Employee Number

  >Base Data Type: 5/555-5           Internal Data Type: I5
                                     Screen Field Width: 7
---- >Input Transforms (Before) --------------------------- Line   1 of   0 ---


---- >Validation (After)  ---------------------------------- Line   1 of   1 ---
Expr: (10 #residue &DATA) #elementof 1 3 5
Mesg: Last Digit is not one of 1,3,5
---- >Output Transforms (After)  ------------------------- Line   1 of   0 ---


---- Validation Example. ------------------------------------------------------
Environment:
Input...: 1/234-4
Internal: **ERROR**: Last Digit is not one of 1,3,5
Output..: 1/234-4
-- 1=Help 2=Send 3=Back 4=Add 5=Del 6=Zoom 7=Up 8=Dn 9=Save 10=Expand 12=Run --
```

Screen 14: Testing "EMPNO" Validation

If the example had been correct, the 'Internal' field would have displayed '12344' - the 'Internal form' of the data. We have finished creating our new Data Type. Press PF3 to return to Screen Design, and press PF3 again to exit from "AUTO" mode (we don't want to make changes to the last three fields). The screen and the fields on it are now defined as we would like.

The next step is to define the COMMANDS which will be available in the screen. Type COMMANDS on the Command Line at the bottom of the screen, and press enter:

```
------------------ System SHAPE.TUTORIAL: Tutorial System --------------------
COPY .PUBLIC.FORM

------------------------------ Screen Design ---------------------------------
 >Name                >Description                    >Fld >Sta >Grp  Min.Size
  MAIN.EMPFORM         Employee Form                   23    1    0   24  80

Entry:
Exit.:
--------------------------- Define Commands [0] ------------ Command   1 of  3

    Command      PFK    >Definition
    HELP           1     .SHAPE.CMDS.GENERAL.HELP
    BACK           3     .SHAPE.CMDS.GENERAL.BACK
    EXIT                 .SHAPE.CMDS.GENERAL.EXIT

---- 1=Help, 2=Send, 3=Exit, 4=Add, 5=Delete, 6=Zoom, 7=Up, 8=Down, 9=Save ----
```

Screen 15: Default Commands

At this site, there are three 'Default Commands" which appear in newly- created screens. Two of them have PF-keys assigned, HELP=PF1 amd BACK=PF3.

You want to use a set of utility commands which are part of the SHAPE Command Library. These Commands: ADD, DELETE, READ, SAVE and SEND have been written to work with simple screens such as this one, where the screen contains data which maps to a single RECORD on a File.

You can COPY the Commands you need commands from the Command Library using the sequence on the Command Line in the previous screen. After pressing Enter, the

screen will look like this:

```
----------------- System SHAPE.TUTORIAL: Tutorial System -------------------


---------------------------- Screen Design [0] -----------------------------
 >Name                >Description                    >Fld >Sta >Grp  Min.Size
  MAIN.EMPFORM         Employee Form                    23    1    0   24  80

Entry: OPENFILE 'tutorial.empdata'
Exit.:
---------------------------- Define Commands ------------ Command   1 of  8

    Command      PFK    >Definition
    HELP           1     HELP
    SEND           2     .PUBLIC.FORM.SEND
    BACK           3     BACK
    ADD            4     .PUBLIC.FORM.ADD
    DELETE         5     .PUBLIC.FORM.DELETE
    SAVE           9     .PUBLIC.FORM.SAVE
    READ          10     .PUBLIC.FORM.READ
    EXIT                 EXIT



---- 1=Help, 2=Send, 3=Exit, 4=Add, 5=Delete, 6=Zoom, 7=Up, 8=Down, 9=Save ----
```

Screen 16: All the Commands you'll need

The fields at the top right of the screen have the following meaning:

Fld: The screen contains 23 fields (the 9 input fields, a /COMMAND, /MESSAGE, /PFLINE, /SYSTEMNAME and /SCREENNAME, plus 9 fixed-text fields (the 'leader' fields for each input field).
Sta: The field has one STATE which is just as well because multi-state screens are not for beginners.
Grp: There are no Field Groups.
Size: The minimun size screen on which the screen can be displayed is 24 rows by 80 columns.

If you want to enter HELP to be displayed when PF1 is pressed, ZOOM (PF6) on the 'Description' field. If you want to take a closer look at the fields, displyed as a Table representing the Format Matrix, ZOOM on 'Fld'. To define field groups, ZOOM on 'Grp' (don't ZOOM on 'Sta')!

The only thing missing now is to type in an 'Entry' expression to open the file you will be using. The expression OPENFILE 'tutorial.empdata' is what you need. Enter it as shown on the screen above, and press PF9 to SAVE the Screen Definition.

You may have wondered about the [0] displayed after the text 'Screen Design' near the top of the screen. This is a Version Number. SHAPE stores old versions of everything. To bring back a previous version of the screen, type the command 'VERSION 1' on the

Command Line (to do this now would bring back the 'empty' screen before we started making any modifications). Type 'VERSION 0' again to bring back the newest version. The command VERSION displays what versions SHAPE still has lying around. To make any previous version the current version, type 'VERSION n' to bring it onto the screen, and press PF9 (save). Don't do this now!

There is only one thing missing before we can run our application: we must define a file in which to store our Data. Type 'FILES' on the command line:

```
------------------ System SHAPE.TUTORIAL: Tutorial System --------------------

-- File     1 of   0 --------------- Files ------------- In Library SHAPE ----

 >File                    Type     >Shr   >Description
  TUTORIAL.EMPDATA        VUPT             Personnel File for Tutorial









-------- 1=Help 2=Send 3=Back 4=Add 5=Delete 6=Zoom 7=Up 8=Down 9=Save --------
```

Screen 17: Files

To start with, there were no files Declared. Type in the line above and press PF9 to add a file declaration for the Symbolic Name 'TUTORIAL.EMPDATA'. Note that this is not the name of a real file, but the name to which SHAPE applications will refer. The actual 'External Name' of the file is defined in the File Design screen, which is the next one we're going to look at.

Move the cursor down to point to our new file and press PF6 (ZOOM):

```
------------------ System SHAPE.TUTORIAL: Tutorial System --------------------

---------- File TUTORIAL.EMPDATA(VUPT) - Personnel File for Tutorial ---------

              External File Names                      Update?
        PROD:
        TEST:

-------- Record Layout ------------------------------------ Field 1 of 0 ----

   Name          Type       Internal    Width  Dummy




-------- 1=Help 2=Send 3=Back 4=Add 5=Delete 6=Zoom 7=Up 8=Down 9=Save --------
```

Screen 18: Nothing Much

There is not much on this screen. Fortunately, we have already designed the fields we want to use in our screen, and we can copy them from there, rather than entering the all the File Definitions to make up the Record Layout. The command 'COPY SCREEN MAIN.EMPFORM' does the trick (we remembered to enter Field Names for all the fields during Screen Design, didn't we?): .

```
------------------ System SHAPE.TUTORIAL: Tutorial System --------------------

---------- File TUTORIAL.EMPDATA(VUPT) - Personnel File for Tutorial ---------

            External File Names                      Update?
        PROD: 1
        TEST:

-------- Record Layout ------------------------------------ Field 1 of 9 ----

   Name          Type        Internal    Width  Dummy
   LASTNAME      CHARACTER   CHARACTER      30  'X'
   FIRSTNAME     CHARACTER   CHARACTER      30  'X'
   PHONENO       555-5555    INTEGER         7  100000+?8999999
   INITIALS      CHARACTER   CHARACTER       6  'X'
   SIN           CHARACTER   CHARACTER      13  'X'
   EMPNO         EMPNO       INTEGER         5  10000+?89999
   ADDRESS       CHARACTER   CHARACTER     150  'X'
   COMMENTS      CHARACTER   CHARACTER     210  'X'



-------- 1=Help 2=Send 3=Back 4=Add 5=Delete 6=Zoom 7=Up 8=Down 9=Save --------
```

Screen 19: The Complete Record Layout

The last thing to be done here is to enter the external file name. For VIEWPOINT the external File Name is actualy a number, which has been entered in the 'PROD' field above.

You may wonder what the entries in the 'DUMMY' field are: these are values which you would have generated, if you declared a TEST file with an external name of 'DUMMY'. This would give you a virtual test file which fed your applcation random numbers or characters, for testing purposes.

If you press PF9 now, you will save the File Definition. However, you still do not have a file. SHAPE does not know whether the file already exists (you could be writing an application to tap into an existing Data Base). You have to type the command CREATE to get SHAPE to attempt to create the file for you.

So... you now have one "finished" application. Unless your site is using special procedures, you can now run the application by loading workspace '1 MENU' and selecting it (it should appeard on the menu you get when you load this workspace). Alternatively, you can type the Command 'RUN TUTORIAL' anywhere within SHAPE. This temporarily suspends SHAPE and enters your application, starting at the Main Menu.